

VNC: A Faster Alternative to X11

Virtual Network Computing (VNC) software provides a way to reduce X11 overhead on high-latency networks such as the Internet. In practical terms, once a VNC session is underway, latencies are on the order of seconds rather than minutes. VNC can make remote X11 applications useful instead of being tedious and non-productive.

The principle of operation involves a host server process (for example, Xvnc) that communicates with X11 applications running on Pleiades. The host server process transmits images and image updates using a low-overhead protocol to the remote system's viewer client.

Security and Firewalls

In the NAS environment VNC traffic is carried by a SSH tunnel, similar to the way SSH is used to tunnel X11 traffic. Using an SSH tunnel provides security because SSH encrypts tunnel traffic in both directions. If you are already using SSH, then VNC traffic will travel to/from NAS systems over current connections and through current firewalls. There is no need for any additional communication updates or authorizations.

Where is the VNC Software?

The Pleiades system runs on Linux. All of the necessary VNC software is installed in `/usr/bin`.

You do not need to run an X11 server on the remote system (your local system) because in the VNC environment, all of the X11 work is done on the Pleiades front-end systems (pfe[20-27]). However, you do need a VNC client viewer. The client might already be installed in many Linux distributions and on recent versions of Mac OS X; if it is not installed on your system, you will need to download the client.

If you have a NAS-supported system, please note that:

- For NAS-supported Linux workstations, a VNC client viewer (RealVNC version 4.1.2) should be installed in `/usr/bin/vncviewer`.
- For NAS-supported Mac workstations, you can download a VNC client called TigerVNC from the [TigerVNC website](#).

If you have a Windows desktop system, you can download free VNC clients from the following websites:

- [Real VNC](#)
- [Tight VNC](#)
- [UVNC](#)
- [TigerVNC](#)

Ask your local system administrator for help to install the VNC client software.

Steps to Establish a VNC Session

In the following steps, pfe24 is used as an example; you can substitute another PFE.

Note: Although there are other ways to establish a VNC session, this method is convenient as it does not require you to manually find an available display number to use.

Before You Begin

VNC is much easier to use if you set up [SSH Passthrough](#) on your local system. In your `.ssh/config` file on your local system, you do not need to enable SSH X11 forwarding, but you must include the line `ForwardAgent yes`.

Known Issue: Make sure you do not have a MATLAB, Tecplot, or FieldView module loaded when you invoke `vncserver`. Once the VNC session is established, you can load the module.

Step 1: Connect to the PFE

Once SSH Passthrough is set up properly, you can establish a SSH connection from your local system to pfe24:

```
your_local_system% ssh pfe24
pfe24%
```

Step 2: Run the vncserver Command on pfe24

`vncserver` is a script that starts/stops/kills the actual VNC server, Xvnc.

The first time you invoke `vncserver` on a server, you will be prompted to create a password for VNC that is up to 8 characters in length. (If you create a longer password, it will be truncated to 8 characters.) This password is encrypted and saved in the `$HOME/.vnc/passwd` file on the server. Once this is done, you will not be prompted for a password on the server when you invoke `vncserver` for subsequent VNC connections.

Run `vncserver` as follows:

```
pfe24% vncserver -localhost

You will require a password to access your desktops.

Password: <--- type in a password of your choice

Warning: password truncated to the length of 8.
Verify: <--- retype your password

New 'X' desktop is pfe24:25

Creating default startup script /u/username/.vnc/xstartup
Starting applications specified in /u/username/.vnc/xstartup
Log file is /u/username/.vnc/pfe24:25.log
```

There are a few options to the `vncserver` command, such as `:display` (for setting the display number), `-geometry` (for setting the desktop width and height in pixel), etc. The `-localhost` option shown in the above example is a local security option that you should use all the time. It must appear as the last option or it won't get processed.

Similar to an X11 session, a VNC session uses a display number. If not supplied, the `vncserver` searches over the valid range from 0 to 99 and assigns the next free display number for your session. In the above example, a display number of 25 is assigned.

Step 3: Create a SSH Tunnel from Your Local System to the Server

The next step is to create a SSH tunnel from your local system to the server. This is done by first escaping into an SSH sub-shell and specifying a local client's port number and a server's port number to use. The default SSH escape characters are `~c` (upper case 'C'). If you do not get the SSH prompt, repeat the `~c`.

```
pfe24% ~C
ssh> -L 59xx:localhost:59xx
Forwarding port.
```

At the SSH prompt, provide a local client port and a remote server port. VNC by default uses TCP port 5900+xx. Thus, it is common to provide the value 59xx for both the local client port (the number before `localhost`) and server port (the number after `localhost`). The value for xx is obtained from the final output from the `vncserver` startup command. In the example shown in Step 2, a `vncserver` was started on pfe24:25, so in this scenario xx would have a value of 25. The port number would therefore be 5925.

Note that the client port number and the server port number do not need to be the same. Some may suggest using a very high client port number such as 22222 or 33333 since high port numbers are less likely to be reserved for other purposes. For example:

```
pfe24% ~C
ssh> -L 22222:localhost:5925
Forwarding port.
```

The maximum number allowed for the client port is 65535. Avoid using the local port numbers 0-1024 (root privilege required), 5900 (for Mac systems, reserved for some Apple remote desktop products), and 6000-6063 (reserved for local X window server). Use the `netstat -an` command to check what local port numbers have been used:

```
your_local_system% netstat -an | less
tcp46      0      0  *.5900          *.*             LISTEN
tcp4       0      0  *.22            *.*             LISTEN
```

The above example shows local ports 5900 and 22 are in use and should be avoided.

Step 4: Start the VNC Viewer Application on Your Local System

- If your local system is a Mac and you have "Chicken of the VNC" installed, launch it. Open the Preferences panel from the "Chicken of the VNC" menu and select the Performance tab. Make sure the "Frontmost Connection" slider is not at its highest setting. If it is, move it down one notch. Close the Preferences panel. Now, open a new connection using the "New Connection" item from the "Connection" menu.

In the popup window "Connect", enter localhost:22222 in the Host field (if your local port number is 22222 from Step 3), and your VNC password in the Password field. Then click on the "Connect" button.

- If your local system is a Linux system, run:

```
your_local_system% vncviewer localhost:localportnumber
```

You should get a password prompt. Enter your VNC password that you created on the server.

The *localportnumber* is the one you use in step 3. For example, if you choose 22222 as your local port, run:

```
your_local_system% vncviewer localhost:22222
```

If everything goes well, the Xvnc server will send a X11 window manager display to your local system that will appear as an xterm in the viewer's window.

The default window manager is TWM, and there are a couple other window managers to choose from in the `/usr/bin` directory, including FVWM, MWM, IceWM, and GNOME. The GNOME window manager provides a GUI view of a user's files and includes a few useful tools.

To use a non-default manager, modify your `$HOME/.vnc/xstartup` file on the host where you start `vncserver`. For example:

```
#twm &
/usr/bin/gnome-session
```

You can also change the size and position of the xterm in your viewer's desktop by changing the values in the following line of the `$HOME/.vnc/xstartup` file on the host where you start `vncserver`. For example:

```
xterm -geometry 80x24+10+10 -ls -title "$VNCDESKTOP Desktop" &
```

This specifies an xterm that is 80 characters wide, 24 characters high, at a position (10 pixels, 10 pixels) from the upper left corner of the VNC viewer's desktop.

TIP: The modifications to the `xstartup` file only take effect for a new VNC connection. You will need to stop the existing VNC server and start a new one.

The window manager's xterm is running on pfe24 itself. From this xterm, you can do tasks that you normally do on pfe24—for example, start an X application or `ssh` to other NAS systems. PBS jobs can also connect to a VNC session. Specifically, in the xterm in the viewer's window, submit an interactive PBS job with the `-X` option (upper case 'X') and do not reset the `DISPLAY` variable before starting an X application:

```
pfe24% qsub - -lselect=1:ncpus=28:model=bro,walltime=1:00:00
qsub: job 1030046.pbspl1.nas.nasa.gov ready
PBS> xclock
```

TIP: Your VNC session and the interactive PBS job will continue to be active even if you disconnect from the Pleiades front end where you started `vncserver`. Assuming the PFE where you started `vncserver` is not down, you can reconnect to the same VNC session: simply `ssh` into the PFE (pfe24 in this example) and repeat steps 3 and 4 with the same port number that you used before (5925 in this example). If the interactive PBS session has not reached its wall time limit, the PBS job will be there waiting.

Step 5: Shut Down the Server When You are Done with the VNC Session

On each VNC server, there are a limited number of VNC sockets available. At the end of a session, be sure to exit the VNC application on your local system so that others can use the sockets. In the terminal window where you started up VNC, use the following command to clean up a few temporary socket files `vncserver` had created.

```
pfe24% vncserver -kill :xx (supply the original display number)
```

For example:

```
pfe24% vncserver -kill :25
Killing Xvnc process ID 3435054
```

WARNING: Don't manually kill `vncserver`. Doing so will leave lock and socket files (for example, `/tmp/.X11-unix/X25`, `$HOME/.vnc/pfe24:25.pid`, etc.) on the server.

TIP: If you get a black screen on your VNC viewer, try the following methods to resolve the issue:

1. Check `/tmp/.X11-unix` for any existing VNC sessions, and clean them up by using the `vncserver -kill :xx` command, as described in Step 5 above.
2. If you normally load MATLAB, Tecplot, or other GUI application modules, unload them before you start `vncserver`. The `LIBGL_ALWAYS_INDIRECT=y` setting in these modules is known to cause the black screen.
3. If unloading the MATLAB and Tecplot modules does not solve the problem, use the `twm` window manager instead of `icewm` or `gnome-session` in your `.vnc/xstartup` file. For unknown reasons, it is possible that after you resolve the black screen issue by using `twm`, you can revert back to using other window managers.

Article ID: 257

Last updated: 18 Jun, 2020

Revision: 51

Logging In -> Setting Up Your Login for Improved Graphics Performance -> VNC: A Faster Alternative to X11
<https://www.nas.nasa.gov/hecc/support/kb/entry/257/>